REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
31-03-2011

2. REPORT TYPE
Proceedings

3. DATES COVERED *(From - To)*
MAR 2011 - APR 2011

4. TITLE AND SUBTITLE
Routing and Rate-Control for Coded Cooperation in a Satellite-Terrestrial Network

5a. CONTRACT NUMBER
FA8720-05-C-0002

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Brooke Shrader, Thomas H. Shake, and Andrew P. Worthen

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
1777 N Kent St, Suite 9030
Arlington, VA 22209

10. SPONSOR/MONITOR'S ACRONYM(S)
OSD-ATL, DDR&E

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
We address the problem of high-throughput, delay-constrained communication over a satellite-terrestrial network where terrestrial node mobility leads to intermittent links. Due to the short time-scale of the link durations in this scenario, standard single-path routing protocols are disadvantaged by the delay incurred In determining that a route is unavailable and then finding a new route. Instead we focus on the approach of sending data over multiple paths simultaneously, and use random linear network coding as a distributed way of sending linearly-independent data on different paths. To ensure efficient use of bandwidth, we present a routing and rate control protocol for coded multipath routing. This protocol specifies the fraction of offered traffic carried on each path, provides a congestion avoidance strategy to limit queueing delays in the network, and adapts quickly to time-varying connectivity. We outline our coded routing and rate-control strategy and also present simulation results from a mobile satellite~terrestrial network.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:
U

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
U	U	U	SAR	6

19a. NAME OF RESPONSIBLE PERSON
Zach Sweet

19b. TELEPHONE NUMBER *(include area code)*
781-981-5997

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

MS-56560

Routing and Rate-Control for Coded Cooperation in a Satellite-Terrestrial Network

Brooke Shrader, Thomas H. Shake, Andrew P. Worth

Abstract—**We address the problem of high-throughput, delay-constrained communication over a satellite-terrestrial network where terrestrial node mobility leads to intermittent links. Due to the short time-scale of the link durations in this scenario, standard single-path routing protocols are disadvantaged by the delay incurred in determining that a route is unavailable and then finding a new route. Instead we focus on the approach of sending data over multiple paths simultaneously, and use random linear network coding as a distributed way of sending linearly-independent data on different paths. To ensure efficient use of bandwidth, we present a routing and rate control protocol for coded multipath routing. This protocol specifies the fraction of offered traffic carried on each path, provides a congestion avoidance strategy to limit queueing delays in the network, and adapts quickly to time-varying connectivity. We outline our coded routing and rate-control strategy and also present simulation results from a mobile satellite-terrestrial network.**

I. Introduction and Problem Setting

In this work we model a mobile satellite-terrestrial network as a time-varying packet erasure network, provide an upper bound on the fraction of transmitted packets that can be received within a dealine, and propose a distributed routing and rate-control strategy that aims to achieve this upper bound. We are motivated by communication in the "urban canyon" scenario, where links are intermittent due to obstructions in the environment. To address the challenges of intermittent links and time-varying topology, we make use of random linear coding to mix and spread packets over multiple paths. Random linear network coding, where random linear combinations of data packets are sent through the network, was originally introduced in [4] and shown to outperform a randomized routing approach in terms of multicast throughput. We use the generation-based random linear coding scheme presented in [2]: data packets arriving at a source node are split into blocks or generations, and the randomly-generated coefficients for each coded packet, or random linear combination over packets in the same generation, is included in the header of the coded packet for later decoding at the destination node. We apply this approach to transmission of packets for a unicast flow. In [3] it was shown that for a wireless packet erasure network, where transmissions are broadcast to all neighboring nodes and transmitted packets are either received without error or erased, a network coding strategy can achieve the max-flow min-cut upper bound on unicast throughput. With this in mind, we design a routing and rate-control strategy that aims to achieve the max-flow, which is a time-varying quantity due to node mobility. This is an extension of our previous work in [10], which addresses a satellite-terrestrial network with fixed, fully-connected terrestrial topology; here we generalize that work by supporting an arbitrary, time-varying terrestrial topology.

Figure 1 summarizes the problem we consider. We wish to send unicast traffic over the downlink of a satellite-terrestrial network in which N terrestrial nodes form an arbitrary topology. A single traffic flow of constant rate λ bps (ie, periodic and deterministic packet arrivals) originates at source node s and is sent over a lossless satellite uplink operating at rate R_S bps, where $\lambda \leq R_S$. The satellite broadcasts the data on the downlink, which also operates at rate R_S, for intended reception at the destination node d. We assume that all N terrestrial nodes fall within the same satellite spotbeam and that transmissions to and from the satellite incur a large propagation delay. Further, we assume that the N terrestrial nodes are mobile vehicles in an urban area that experience satellite blockage due to obstructions by buildings, trees, and other objects. We use two different approaches to modeling the time-varying satellite blockage: data from field measurements and a Markov model. For the field measurement data, we use the blockage realizations from the measurement campaign described in [11], where it was shown that the received signal-strength has an on-off behavior that is well-modeled by a channel with two states: blocked and unblocked. A vehicle in the blocked state does not receive any data transmitted by the satellite and a vehicle in the unblocked state receives all data sent by the satellite without error. Data from the measurement campaign have been fit with a two-state Markov model, which we also use. More details about the models we use are provided in Section IV.

When the link from the satellite to d is blocked, the other terrestrial nodes can cooperate to relay data to d. We assume a range-based model for terrestrial links: a pair of terrestrial nodes separated by a distance of at most r form a bi-directional link. We assume that transmissions by terrestrial nodes are broadcast transmissions that can be received by all nodes located within distance r of the transmitting node. We let R_n denote the bps transmission rate for all data sent by terrestrial node n. We assume that terrestrial links do not incur losses, but allow the possibility that some packets can be lost or dropped at a terrestrial node due to queue buffer overflow. The buffer overflow and associated packet dropping problem is particularly relevant when $R_n < R_S$; this is the case we consider in much of our work. Our objective is to maximize the

The authors are with the Massachusetts Institute of Technology (MIT) Lincoln Laboratory, Lexington MA. This work was sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

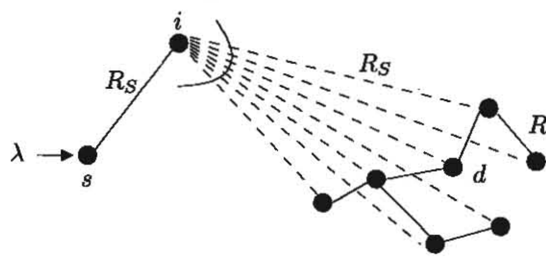


Fig. 1. Network setting under consideration. Data is to be sent from source s to destination d via satellite i. The N terrestrial nodes form an arbitrary and potentially time-varying topology. In this figure $N = 7$.

throughput from s to d while meeting a delay constraint; we would like to achieve this objective over a wide range of values of offered load λ. Note that the heterogeneity in link rates R_S and R_n can lead to congestion, which is problematic for meeting a delay constraint. Due to the mobility of terrestrial nodes and the associated satellite blockage, the amount of data that the network can support for reception at d is a time-varying quantity. Thus the routing of packets through the terrestrial network must adapt to mobility, and the rate that data is sent on each terrestrial link must be allocated appropriately.

II. Approach and Performance Prediction

We can treat the routing and rate-control problem outlined above as a time-varying maximum-flow problem. Let $G(t) = (V(t), E(t))$ denote the flow network corresponding to the satellite-terrestrial network described above at time t. The weighted graph $G(t)$ is similar to the connectivity graph in our satellite-terrestrial network, where there is an edge between the satellite and terrestrial node n if n is unblocked at time t. In the flow network, we add "dummy" vertices and corresponding edges to represent broadcast transmissions on the satellite downlink and terrestrial links. Specifically, suppose a node v in our satellite-terrestrial network broadcasts at rate R to nodes u and w. Then to transform the connectivity graph of our satellite-terrestrial network to our flow network $G(t)$, we remove edges (v, u) and (v, w), add a "dummy" vertex v', and add edges (v, v'), (v', u), and (v', w), each with weight R.

Let $C(t)$ denote the maximum-flow for the network $G(t)$. We compute $C(t)$ using the Edmonds-Karp algorithm [13], a centralized algorithm that provides us an estimate of the maximum amount of data in bps that can be delivered from s to d at time t. We use $C(t)$ to benchmark the maximum throughput we aim to achieve at time t through a distributed routing and rate-control protocol. Our aim is at time t to achieve a throughput of $\min(\lambda, C(t))$ bps. Note that this implies that if the rate of offered traffic λ is larger than the max-flow $C(t)$ that the network can instantaneously support, then a fraction $(\lambda - C(t))/\lambda$ of the offered traffic will not be delivered to the destination. Additionally, we use $C(t)$ to compute a benchmark for the time-average performance of various distributed protocols. We are interested in the metric of packet completion rate, which is computed as the fraction of data packets generated at the source that are received at d within a deadline. We can compute the max-flow of the network at T regularly spaced intervals of period t seconds and then compute an upper bound on the packet completion rate as

$$S = \frac{1}{\lambda t T} \sum_{k=1}^{T} \min(\lambda, C(kt)). \qquad (1)$$

As described more extensively in later sections, we consider scenarios in which the number of nodes N and the link rates R_n are time-invariant; yet, because of link blockages, at certain times $C(t) > \lambda$, while at other times (perhaps only a short period later), $C(t) < \lambda$. Our aim is to implement a routing and rate-control scheme that adapts to the time-variability of the max-flow, while also adapting to the fact that the offered traffic rate λ, which we assume to be time-invariant, may not always be feasible or supportable by the network.

Our approach involves allocating routes and the rates at which traffic is served on those routes so that data is delivered to d at a rate of at most $C(t)$, and if $\lambda > C(t)$, we actively discard some of the offered traffic at intermediate or relay nodes in the network. The motivation for this approach comes from specific features of the satellite-terrestrial network under consideration. In a time-invariant or slowly-varying network, a standard and appropriate approach is to perform flow-control so that if the rate of offered traffic overburdens the network, then the offered traffic rate is reduced *at the source* node, and potentially increased again if or when the network can support it. In a time-varying network, flow control can also be applied, and its effectiveness will depend on how often the offered traffic rate is adjusted at the source relative to how often the rate that the network can support, in this case given by the max-flow $C(t)$, changes. In our satellite-terrestrial network, the propagation delay to and from the satellite inhibits the ability of the source node to quickly adapt the rate of offered traffic. The propagation delay is a large fraction of a second, while the max-flow $C(t)$ can change on the order of every second; thus flow control may not be very effective. Yet another approach is to form an estimate of the time-average of the max-flow $C(t)$ and to inject packets at the source at that rate; however this approach will not prevent congestion and the associated queueing delay at intermediate nodes when the instantaneous max-flow $C(t)$ is smaller than its time average.

III. Routing and Rate-Control Protocol

A central feature of our protocol is random linear network coding, which enables efficient multi-path routing by limiting the amount of redundant data carried on different paths. We use the generation-based random linear network coding scheme outlined in [2]: packets arriving at the source node are grouped into blocks or generations consisting of K data packets each. The source marks each generation of packets with a sequence number, which is included in all transmitted packets. A coded packet is formed by a linear combination of K packets from the same generation, where the coefficients of the linear combination are chosen randomly and uniformly from a large finite field; the set of coefficients, termed the

encoding vector, is appended to each coded packet that is sent. The destination node d collects coded packets, and if it receives a full-rank matrix of encoding vectors from the same generation, it performs Gaussian elimination to recover the original data packets for that generation. Multiple generations propagate through the network concurrently. Each intermediate node can maintain a finite number of generations in memory and removes older generations (as determined by sequence number) when its memory becomes full.

Below we describe a routing and rate-control protocol that is used at terrestrial nodes in the network. This protocol is not used at the source or satellite nodes; instead those nodes simply transmit one packet for each packet they receive. As described below, terrestrial nodes exchange separate control packets used to find routes and allocate rates. Among other functions, control packets allow nodes to estimate their distance DIST in hops from the destination and this value is included in all control and data packets sent through the network. The value of DIST can be easily discovered and maintained: the destination node sets its DIST to zero, and the other terrestrial nodes set their DIST to one plus the smallest value of DIST that they have recently received from neighboring nodes. The DIST value allows each node to have a notion of which neighboring nodes are upstream, or further from the destination, and which are downstream, or closer to the destination. This information is used for routing and rate control. Also each node v maintains a list $\mathcal{U}(v)$ of its 1-hop upstream neighboring nodes.

A. Estimating max-flow at the destination

Control packets are used to allow the destination node d to form an estimate of the time-varying max-flow rate $C(t)$. Each non-destination node v will send control messages with an advertised rate A_v representing the bps rate that node v believes it can contribute to sending traffic for the flow. The one-hop downstream neighbors of v that receive this control message will store the value of A_v with their upstream neighbor list and use it in computing their own advertised rates. Note that the value of A_v is not explicitly propagated further than one hop. Node v calculates its advertised rate as follows.

$$A_v = \begin{cases} R_v, & v \text{ unblocked} \\ \min\left(R_v, \sum_{n\in\mathcal{U}(v)} A_n\right), & v \text{ blocked} \end{cases} \tag{2}$$

This value is updated and a new control packet is sent whenever (1) a node's satellite link changes state (blocked/unblocked) or (2) a node receives a new advertised rate from upstream, and this new information changes the nodes' own advertised rate. The destination node d estimates the max-flow using this information. We let $\hat{C}(t)$ denote the destination nodes' estimate of max-flow, which is computed as the sum $\sum_{v\in\mathcal{U}(d)} A_v$.

B. Rate control

Intermediate nodes in the network transmit coded packets in response to requests originating at the destination node d when d becomes blocked. Likewise, when the satellite link state at d transitions from the blocked to unblocked state, it sends a control message upstream requesting that intermediate nodes stop sending coded packets. In response to requests originating at the destination, intermediate nodes contruct rules for whether and how many coded packets to send. The intermediate node applies these rules to all generations for which it receives coded packets after the request is received. When a new request is received, a new set of rules are constructed and applied. The set of rules discussed here has two components: a generation discard strategy for congestion avoidance and an allocation of packets per generation to be transmitted downstream by the node.

The generation discard strategy operates as follows. The destination d forms as estimate of the offered traffic rate λ, either by estimating λ when it is unblocked or gathering estimates from intermediate nodes. Here we assume that d has a perfect estimate of λ, which is reasonable since we also assume that λ is the rate of a deterministic, time-invariant process. The destination node then estimates the fraction γ of incoming traffic that can be served by the network as follows.

$$\gamma = \min\left(1, \frac{\hat{C}(t)}{\lambda}\right) \tag{3}$$

The value of γ is included in control messages that the destination node sends to request coded packets, and nodes that receive the requests copy the same value of γ into any requests sent upstream. All nodes in the network are provided at initialization the same integer value baseNS, which is chosen to be large so that $\gamma \times \mathsf{baseNS}$ is close to an integer value. Then for each newly arriving generation with sequence number GenID that arrives at an unblocked intermediate node, the following operation is performed.

if $(\mathsf{GenID} \times \lfloor\gamma\mathsf{baseNS}\rfloor) \bmod \mathsf{baseNS} < \lfloor\gamma\mathsf{baseNS}\rfloor$ **then**
  send coded packets for generation GenID
else
  discard generation GenID
end if

This provides a distributed and deterministic way for intermediate nodes to identify the *same* generations to be discarded. Note that if different intermediate nodes discard packets from different generations, then the destination may receive many incomplete generations that cannot be decoded; this is the reason that intermediate nodes must coordinate to discard the same generations.

When sending coded packets for a generation, each node determines how many coded packets to send using information from downstream nodes. When a request for coded packets, originating at the destination and propagating upstream, is sent by node v, node v specifies the number of packets per generation being requested, given by PPG_v and the sum of the advertised rates in the upstream neighbor list of v. Then a node u that is upstream from v and receives this request will determine how many coded packets to send for each generation

as follows.

$$\mathsf{PPG}_u = \mathsf{PPG}_v \frac{A_u}{\sum_{n \in \mathcal{U}(v)} A_n} \quad (4)$$

If node u is unblocked, it will receive packets from the satellite and will transmit PPG_u coded packets as soon as it receives a complete generation; in this case node u will not send any further requests upstream. If node u is blocked, it will send a request for PPG_u packets upstream, and will transmit at most PPG_u coded packets after receiving from its upstream neighbors. The destination node d will always request $\mathsf{PPG}_d = K$ coded packets for each generation. Intermediate nodes can also adjust their rate allocation in response to changes in link state that cause the advertised rate A_v to change. Finally, note that PPG_u may be a non-integer value; if so, the node will transmit $\lfloor \mathsf{PPG}_u \rfloor$ coded packets, plus one extra coded packet with probability $\mathsf{PPG}_u - \lfloor \mathsf{PPG}_u \rfloor$.

C. Repair traffic

Our routing and rate control scheme also includes a strategy for repairing incomplete generations at the destination node. Specifically, the rate control strategy outlined above aims to discard (if necessary) a fraction of generations that cannot be served given the current state of the network and to split the rate that intermediate nodes send traffic by dividing up the number of packets per generation that the destination node needs. However, it is possible that for certain generations, the destination will receive fewer than K packets and will be unable to decode; this may be caused by propagation delays for control messages, link state changes which cause the rate allocation to be modified, or the probabilistic rule for determining number of coded packets to send when PPG_u is non-integer. To make the rate control scheme robust to these time-delays, state changes, and random errors, we use a request-based generation-repair mechanism whereby additional packets are requested to ensure a complete generation. Specifically, the destination node periodically checks for incomplete generations of coded packets and sends control messages specifying the GenID and rank-deficiency of a batch of incomplete generations. The intermediate nodes answer those requests by sending additional coded packets for the specified generations, using the same rate-splitting strategy as shown above for determining PPG_u. Coded packets sent in response to a repair request are given higher priority over other coded packets. Moreover, if the volume of coded packets sent as repair response is large, it will impact the ability of the network to send newly-arriving generations. For this reason, the destination must account for repair traffic in estimating the fraction of incoming traffic that can be served. Let ω denote the estimate formed at d of the bps rate that coded packets must be sent for repair requests; node d can estimate ω by measuring the rate that it receives packets from incomplete generations. Then the destination will estimate

$$\gamma = \min\left(1, \frac{\hat{C}(t)}{\lambda + \omega}\right) \quad (5)$$

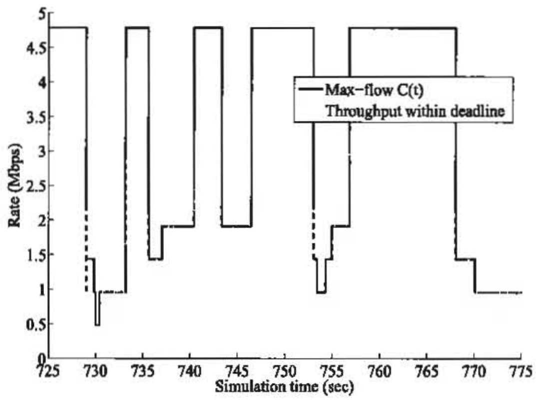
and use this value for the generation discard strategy.

IV. Experimental results

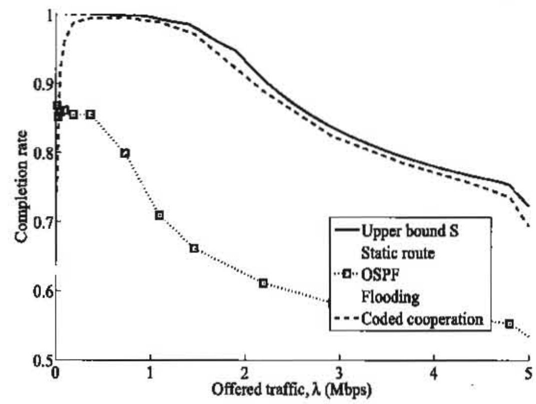
The routing and rate control strategy outlined above has been implemented and tested in simulation. Application data arrives at the source through periodic arrivals of fixed-length packets consisting of 1400 bytes and are sent over UDP, which prepends a 28-byte header. The source and satellite nodes have an output queue for transmission that allow for buffering at most 100 packets and the one-way propagation delay for the satellite is 125 msec (i.e., 250 msec total for the uplink and downlink). Intermediate nodes can buffer at most 50 packets in an output queue for transmission and packets must be received at the destination within a deadline of 4 seconds after their arrival at the source. We set link rates as $R_S = 5$ Mbps and $R_n = 500$ kbps for all terrestrial nodes. Network coding is implemented as an IP overlay in which node s is the overlay ingress node, node d is the overlay egress node, and all other nodes participate in the overlay throughout the entire simulation. We prepend an additional $52 + K$ byte header on network coding packets; the header specifies encoding vectors, GenID values, DIST values, and other control information. Every node in the network can store in memory packets for at most 100 generations at a time; if the memory is full and a packet for a new generation arrives, the oldest generation is flushed, or dropped. Network coding is performed over the finite field $GF(2^8)$ with generation size $K = 8$.

We test three different baseline protocols for performance comparison. Two of the baseline protocols are single-path routing strategies: static routing and the Open Shortest Path First (OSPF) protocol. Static routing refers to a fixed route from the satellite to d, so under this strategy none of the other terrestrial relay nodes transmit or assist in any other way. The OSPF protocol implements an adaptive single-path routing that aims to adjust the route as link states change. We tested the OSPFv3 protocol and modified some protocol parameters to allow the routes to adapt more quickly [5]. We set the OSPF timers as follows: Hello Interval 1.0 sec, Dead Interval 2.0 sec, Interface Transmission Delay 0.25 sec, Retransmission Interval 2.0 sec, and SPF Calculation Delay and Hold Time 0 sec. Finally, we tested multiple-path flooding as a baseline protocol. Under this strategy, all terrestrial nodes can retransmit all packets they receive from the satellite or from other terrestrial nodes. Each terrestrial node keeps a list of 5000 packets it previously transmitted, and before transmitting a packet, it checks this list to ensure that it has not recently sent the same packet. This strategy aims to avoid repeated transmission of the same packet by the same node, however, it does not prevent multiple nodes that are equidistant (in hops) to the destination from sending the same packet.

The first set of results shown here are for $N = 9$ terrestrial nodes arranged in a time-invariant X topology with the destination node d at the center and for satellite blockage following the "Boston" two-state Markov model in [11]. Under this blockage model, the two-state Markov chain evolves in discrete time at intervals of 0.1 sec and the self-transition probabilities are 0.9919 in the unblocked state and 0.9866 in the



(a) Max-flow and throughput within deadline versus simulation time



(b) Packet completion rate versus offered load.

Fig. 2. Results for the time-invariant X topology. (a) Max-flow rate $C(t)$ and throughput within deadline for the coded routing and rate control strategy as a function of simulation time, with offered load λ 2.2 Mbps. (b) Packet completion rate versus offered load for static routing, OSPF, multipath flooding, and coded multipath routing. The upper bound on packet completion rate S is also shown.

blocked state. As shown in Fig. 2(a), the max-flow rate $C(t)$ varies rapidly, at times changing as often as once per second, and the coded multipath routing strategy adapts to achieve a throughput near $\min(\lambda, C(t))$. The spikes in throughput shown in this figure appear due to plotting throughput in 0.25 sec bins, batch decoding of packets in the same generation, and simultaneous reception of packets from other terrestrial nodes and from the satellite immediately after d transitions from the blocked to unblocked state. In Fig. 2(b) we show an upper bound on the packet completion rate as well as the performance for the four different strategies. As it is designed to do, the coded multipath routing strategy nearly achieves the upper bound on completion rate given by the time-averaged max-flow. Multipath flooding performs nearly as well, but is hindered by duplicate packets sent on multiple paths. The OSPF protocol is unable to adapt quickly enough to the time-varying link states, while the static routing strategy performs poorly as it fails to implement any form of terrestrial relaying.

We also present results here for a time-varying topology with $N = 7$ mobile terrestrial nodes and satellite blockage realizations from the measurement campaign in Cambridge in [11]. Mobile terrestrial nodes follow the Cambridge routes shown in [11] over an area of roughly 1400 by 1400 meters; nodes are assumed to have a transmission/reception range of $r = 700m$ and each node passes through 3300 different (discrete) locations during the course of the simulation. In Fig. 3 we show the performance for mobile node speed of 7 m/sec (approximately 15 mph). Figure 3(a) shows that the packet completion rate performance of the different protocols follows the same trends as observed for the time-invariant topology. Here, the performance of the coded multipath routing strategy is further from the theoretical upper bound on packet completion rate; this is due to the incidence of non-disjoint paths in the routing subgraph and to the time-varying topology. In Fig. 3(b) we show results for a metric of protocol efficiency which is computed by counting up the total unique number

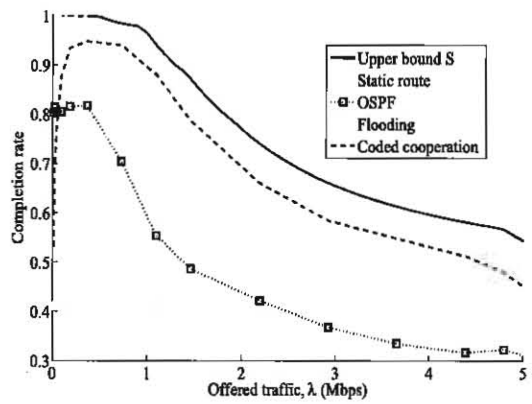
TABLE I
COMPLETION RATE PERFORMANCE AT OFFERED LOAD $\lambda = 366$ KBPS FOR DIFFERENT TERRESTRIAL NODE SPEEDS

Speed (m/sec)	$\overline{T}$ (sec)	Coded Routing	OSPF	Flooding
1.4	6.7	0.98	0.92	0.99
7	1.3	0.95	0.82	0.99
21	0.4	0.90	0.74	0.99

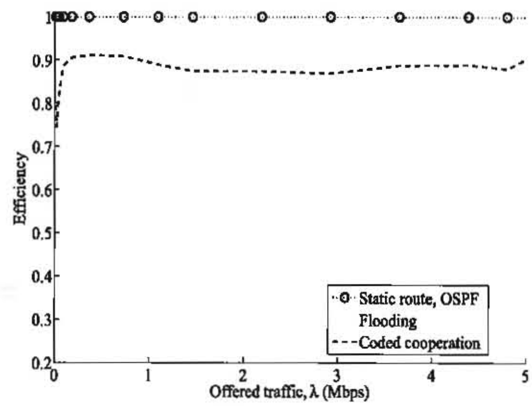
of data packets received and decoded within the deadline at d and dividing by the total number of data packets (coded or uncoded) received at d. Under this metric, the single-path routing strategies obtain an efficiency of 1, since they never send a duplicate packet along any path. The multipath flooding strategy obtains a poor efficiency performance due to its tendency to send duplicate packets along multiple paths. The coded multipath routing strategy is penalized for incomplete generations and for more than K coded packets received at the destination, yet its efficiency performance is good, nearly that of single-path routing. Finally, packet completion rate results for the same Cambridge scenario under three different node speeds is shown in Table I. In addition to node speed, we display the average time $\overline{T}$ over which the max-flow $C(t)$ remains constant. As node speeds increases, blockage durations and the durations of terrestrial node pairs being within range to form a link both become shorter and $\overline{T}$ decreases. The multipath flooding strategy provides the best completion rate performance with increasing node speed; this comes at a cost of redundant packets and inefficient use of bandwidth as shown in Fig. 3(b). On the other hand, the packet completion rate of OSPF scales poorly with increased node speed. The coded multipath routing performs in between these two extremes.

V. DISCUSSION AND RELATED WORK

The design of distributed protocols for multipath routing with network coding has been addressed in numerous previous



(a) Packet completion rate versus offered load



(b) Protocol efficiency versus offered load.

Fig. 3. Comparison of static routing, OSPF, multipath flooding, and coded multipath routing for the Cambridge scenario with mobile node speed 7 m/sec.

works. In [9], the authors present the CodeCast protocol, which is a distributed implementation of the subgraph selection strategies proposed in [8]. CodeCast achieves low packet loss and low latency through careful design of the number and time that each node injects coded packets into the network; however, in constrast to our work, the CodeCast protocol is not designed to handle heterogeneous link rates and does not provide congestion avoidance mechanisms. The MORE protocol proposed in [1] and used in [6] and [7] is a reliable file transfer protocol designed to minimize the number of coded packet transmissions that nodes must make in order to deliver packets to the destination. By contrast, we take a best-effort approach to support delay-constrained transmissions, i.e., we may discard some packets and thus do not guarantee reliability; furthermore, our approach is to maximize the rate at which data is received at the destination. In [12], the authors present the Optimized Multipath Network Coding (OMNC) protocol, which performs routing and rate control and is designed to maximize the rate at which data is delivered to the destination. In contrast to our work, OMNC is designed to support random but time-invariant or slowly-varying link losses. Also OMNC avoids congestion through control of the packet injection rate at the source node, whereas we have avoided this approach due to the satellite propatation delay.

The coded routing and rate control strategies outlined here can provide good performance in satellite-terrestrial networks relevant to practical scenarios, and yet there are multiple avenues for future work. First, we have assumed that terrestrial links are lossless, which will not be a reasonable assumption in all settings. While we expect that the repair mechanisms we propose can help with random terrestrial packet losses, more proactive strategies, such as estimating the terrestrial packet loss rate and accounting for it in computing the max-flow, may perform better. Also, the problem we discuss here is confined to communication over the satellite downlink, and strategies to handle the satellite uplink problem are also necessary. Finally, our routing and rate control strategy is developed for a single unicast flow, and techniques to handle multiple flows are a subject of future research.

ACKNOWLEDGMENT

The authors would like to thank ...

REFERENCES

[1] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *Proc. ACM SIGCOMM,* August 2007.

[2] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," *Proc. Allerton Conference,* 2003.

[3] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Transactions on Information Theory,* vol. 52, no. 3, pp. 789-804, March 2006.

[4] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," *Proc. IEEE International Symposium on Information Theory,* 2003.

[5] D. Kiwior, E. G. Idhaw, and S. V. Pizzi, "Quality of service (QoS) sensitivity for the OSPF protocol in the airborne networking environment," *Proc. IEEE Military Communications Conference,* (MILCOM), 2005.

[6] G. Lauer and D. S. Morris, "Network coding performance: an emulation experiment," *Proc. IEEE Military Communications Conference* (MILCOM), 2008.

[7] V. Firoiu, G. Lauer, B. DeCleene, and S. Nanda, "Experiences with network coding within MANET field experiments," *Proc. IEEE Military Communications Conference* (MILCOM), 2010.

[8] T. Ho and D. S. Lun, *Network Coding: An Introduction,* Cambridge University Press, Cambridge, 2008.

[9] J.-S. Park, D. S. Lun, Y. Yi, M. Gerla, and M. Medard, "CodeCast: a network coding based ad hoc multicast protocol," *IEEE Wireless Communications Magazine,* October 2006.

[10] B. Shrader, A. Babikyan, N. M. Jones, T. Shake, and A. P. Worthen, "Rate control for network coded multipath relaying with time-varying connectivity," *IEEE Journal on Selected Areas in Communication (JSAC), Special Issue on Trading Rate for Delay at the Transport and Application Layers,* vol. 29, no. 5, May 2011.

[11] W. M. Smith, "Channel characterization for EHF satellite communications on the move," *MIT Lincoln Laboratory Technical Report,* TR-1109, 12 July 2006.

[12] X. Zhang and B. Li, "Optimized multipath network coding in lossy wireless networks," *IEEE Journal on Selected Areas in Communications, Special Issue on Network Coding for Wireless Communication Networks,* vol. 27, no. 5, pp. 622-634, June 2009.

[13] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms,* MIT Press, Cambridge, 1990.